УДК 681.323

В.М. ГУСЯТИН, канд. техн. наук, ХНУРЭ,
Я.В. ЧАГОВЕЦ, канд. техн. наук, ХНУРЭ,
Д.Г. КОЖУШКО, ХНУРЭ

УСТРАНЕНИЕ АЛИАСИНГА ПРИ НАНЕСЕНИИ ТЕКСТУРЫ НА 3*D*-ОБЪЕКТЫ В МЕТОДЕ ОБРАТНОГО ТРАССИРОВАНИЯ

Пропонується метод нанесення векторних та растрових текстур з одночасним усуненням аліасінгу при синтезі зображення методом зворотного трасування. Враховується форма проекції пікселя на площину текстури. Суть методу полягає в знаходженні інтегральної суми кольорів, що попали в середину області сумування. Область сумування є апроксимацією реальної форми проекції пікселя на площину текстури. Розглянуті питання підвищення ефективності знаходження інтегрального кольору проекції пікселя.

The method of vector and raster textures mapping is offered with the simultaneous removal of aliasing at the synthesis of image of the ray tracing a method. It is suggested to take into account the form of projection of pixel on the texture plane. Essence of method consists in finding of integral sum of colors which got in the middle of area of sad. An area of sad is approximation of the real form of pixel projection pixel on the texture plane. The questions of increase of efficiency of finding of integral color of projection of pixel are considered.

Постановка проблемы. В процессе синтеза 3*D*-сцен возникает необходимость оперировать с большими объемами текстур. Кроме того, при отображении текстур возникает проблема устранения алиасинга. Предлагается разработать методы нанесения векторных и растровых текстур с одновременным устранением алиасинга при синтезе изображений методом обратного трассирования.

Анализ литературы. В работах [1 – 4] используется метод, основанный на использовании MIP-карт, применение которых позволяет вводить уровни детализации текстуры с целью устранения алиасинга. Недостатком данного подхода является размытие деталей рисунка текстуры при визуализации. В работе [5] предлагается учитывать форму проекции пикселя на плоскость текстуры. Недостатком описаного алгоритма являются значительные вычислительные затраты, кроме того, алгоритм не учитывает все возможные формы проекции пикселя. В статьях [6, 7] предлагается подход к отображению текстур с высоким разрешением при визуализации больших пространств. Недостатком данного подхода является невозможность нанесения текстуры на аналитически заданную поверхность. В статье [8] предлагается формат представления растровой текстуры, который позволяет значительно сократить вычислительные затраты при нахождении интегрального цвета прямоугольной аппроксимации проекции пикселя. К недостаткам можно отнести: возможность применения только для растровых текстур, значительное увеличение объема данных текстуры, грубая аппроксимация проекции пикселя. В статье [9] предлагается подход к устранению алиасинга процедурно-заданных текстур. Недостатком данного похода является

ограниченность его применения узким классом нефотореалистических изображений текстур. В статье [10] авторов предлагаются древовидные форматы упаковки векторных текстур, которые, с одной стороны, позволяют уменьшить объемы памяти, а с другой стороны, позволяют отображать текстуры без предварительной распаковки в реальном масштабе времени. Данные форматы текстур ориентированы на метод устранения алиасинга, рассмотренный в данной статье.

Цель статьи. Разработка метода нанесения текстур с одновременным устранением алиасинга. Данная статья является продолжением [10]. Предлагается учитывать форму пикселя путем суммирования цветов текстур, попавших в так называемую область суммирования. Область суммирования – область на текстуре, являющаяся аппроксимацией реальной формы проекции пикселя на эту текстуру.

Описание метода нанесения текстур. Сцена задана базовой системой координат (с/к) XYZ. Задан наблюдатель (центр проекции точка *h*), с которым связана система координат UVW. Сцена состоит из объектов (рис. 1, а) в с/к $X_iY_iZ_i$ ($i = \overline{1,N}$, N – количество объектов в сцене).

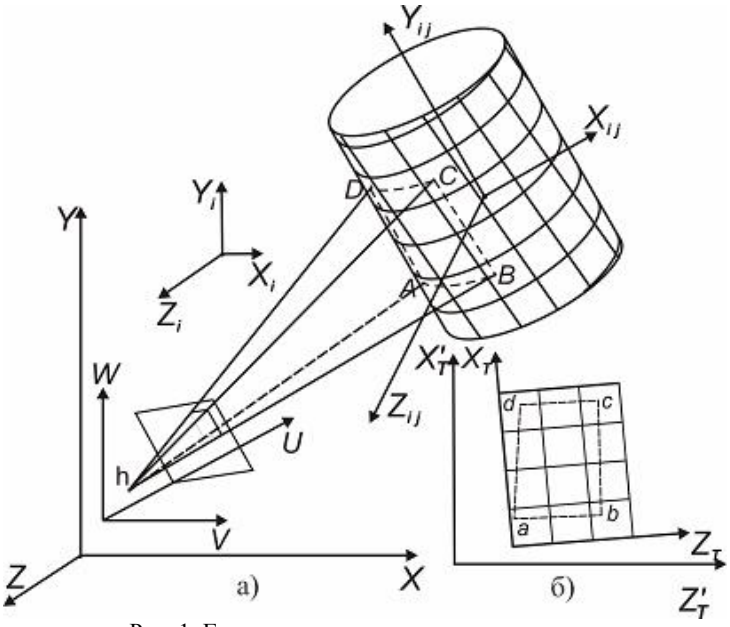

Рис. 1. Геометрические элементы задачи

В свою очередь, i-й объект состоит из примитивов, заданных в с/к $X_{ij}Y_{ij}Z_{ij}$ ($j = \overline{1, K_i}$, K_i – количество примитивов в i-м объекте), на каждый из которых нанесена текстура (рис. 1, а).

Текстура задана на плоскости. Введем функцию, которая устанавливает соответствие между каждой точкой поверхности примитива и точкой на плоскости текстуры:

$$P_T = F(P_{ij}), \tag{1}$$

где $P_{ij} = \{X_{ij}, Y_{ij}, Z_{ij}\}$ – трехмерные координаты точки пересечения проекционного луча с j-м примитивом i-го объекта в с/к $X_{ij}Y_{ij}Z_{ij}$. Например, точки A', B', C', D' (рис. 1, б).

$P_T = \{X_T, Z_T\}$ – двухмерные координаты точки на плоскости текстуры в с/к $X_T Z_T$. Например, точки a, b, c, d (рис. 1, б).

Функция F задается:

$$F(P_{ij}) = F_2(F_1(P_{ij})), \tag{2}$$

где $F_1 : P_{ij} \to P_T^{'}$ – функция нанесения текстуры, перевода трехмерных координат точки, заданной в с/к примитива, в двухмерные на плоскости текстуры в с/к $X_T^{'} Z_T^{'}$;

$F_2 : P_T^{'} \to P_T$ – функция аффинного преобразования в с/к $X_T Z_T$.

Преобразование F_1 используется для отображения точки в трехмерном пространстве на двухмерную плоскость. Примерами функций отображения являются: цилиндрическая, сферическая, линейная. В общем случае с/к $X_T^{'} Z_T^{'}$ не совпадает с $X_T Z_T$. Аффинное преобразование F_2 учитывает масштаб, линейное и угловое смещение текстуры на плоскости.

При синтезе изображения трехмерной сцены методом обратного трассирования из центра проекций h будем трассировать лучи, проходящие через каждую вершину пикселей экрана. Для экрана с разрешением m на n пикселей необходимо провести $(m+1)\times(n+1)$ лучей.

Таким образом, для каждого пикселя рассматривается четверка лучей, проходящих через его вершины. Рассмотрим случай, когда все четыре пересечения проекционного луча принадлежат одному примитиву (рис. 1, а).

В общем случае истинная проекция пикселя на поверхность примитива представляет собой пространственную фигуру (рис. 1, а). Нахождение проекции такой пространственной фигуры на плоскость текстуры представляет собой нетривиальную задачу.

Предлагается находить лишь проекции вершин пикселя на плоскость текстуры. Для этого необходимо выполнить такие этапы:

1. Найти точки пересечения проекционных лучей, проходящих через вершины пикселя, с поверхностью примитива, заданного в с/к $X_{ij}Y_{ij}Z_{ij}$. Полученные точки обозначить A, B, C, D (рис. 1, а).

2. Найти проекцию точек A, B, C, D на плоскость текстуры в с/к $X_T^{'}Z_T^{'}$ – функция F_1. Полученные точки обозначить a', b', c', d'.

3. Перевести двухмерные координаты точек (a', b', c', d') в с/к $X_T Z_T$ путем выполнения аффинного преобразования – функция F_2. Полученные точки обозначить a, b, c, d (рис. 1, б).

Полученный в результате выполнения действий 1 – 3 четырехугольник $abcd$ предлагается рассматривать в качестве аппроксимации истинной проекции пикселя на плоскость текстуры.

Разбиение аппроксимированной проекции пикселя на фрагменты. В общем случае аппроксимированная проекция пикселя (в дальнейшем по тексту просто проекция пикселя (ПП)) представляет собой произвольный четырехугольник $abcd$. Нахождение его интегрального цвета требует значительных вычислительных затрат, обусловленных нахождением пересечения классификационных квадратов (КК) [10] подготовленных текстур с проекцией пикселя. С целью упрощения вычисления интегрального цвета предлагается аппроксимировать четырехугольник $abcd$ набором прямоугольников, стороны которых параллельны осям системы координат текстуры. Данные прямоугольники будем называть фрагментами.

Для составления алгоритма разбиения проекции пикселя на фрагменты предлагается руководствоваться следующими принципами:

1. Совокупность фрагментов полностью покрывает проекцию пикселя.
2. Не допускается перекрытие фрагментов.
3. Количество фрагментов и их размеры выбираются исходя из требуемой точности нахождения интегрального цвета проекции пикселя.

Точность аппроксимации оценивается отношением суммарной площади фрагментов к площади $abcd$. С учетом п.1 данное отношение всегда больше либо равно единице. Минимизация этого отношения позволяет получить более точную аппроксимацию.

Ниже предлагаются основные этапы алгоритма разбиения проекции пикселя на фрагменты.

1. Выбрать значение относительной погрешности аппроксимации ПП δ (см. ниже по тексту).

2. Вычислить уровень k дерева, на котором выполняется разбиение пикселя на фрагменты (см. ниже по тексту).

3. Найти точки p_j^x и p_i^z пересечения сторон четырехугольника $abcd$ с линиями регулярной решетки КК, параллельными осям OX_T и OZ_T

соответственно. При этом $j = \overline{1,n}$; $i = \overline{1,m}$, где n, m – количество точек пересечения с линиями, параллельными осям OX_T и OZ_T соответственно.

4. Из полученных точек пересечения и вершин (a, b, c, d) сформировать два списка: $Lx = \{a, b, c, d, p_1^x, p_2^x, ..., p_n^x\}$, $Lz = \{a, b, c, d, p_1^z, p_2^z, ..., p_m^z\}$.

Например для (рис. 2, а): $Lx = \{a, b, c, d, p_1^x, p_2^x, ..., p_8^x\}$, $Lz = \{a, b, c, d, p_1^z, p_2^z, ..., p_{14}^z\}$.

5. Отсортировать точки списков Lx и Lz в порядке возрастания значения координаты X_T и Z_T соответственно.

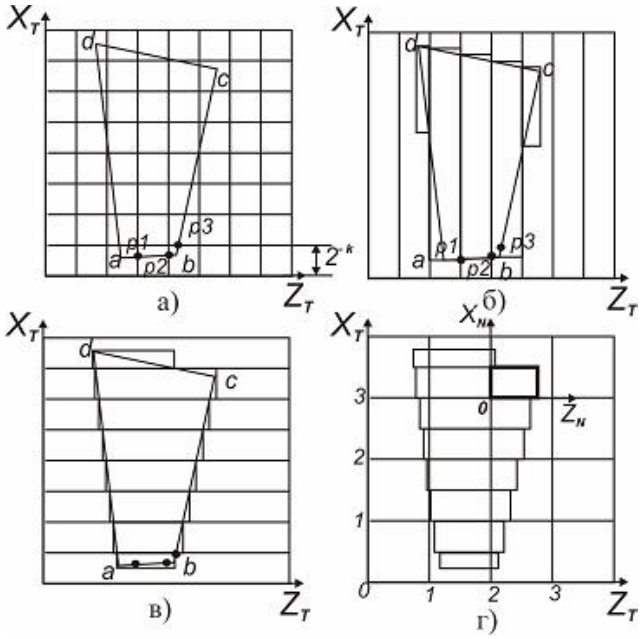

Рис. 2. Разбиение ПП на фрагменты

6. Из полученных списков Lx и Lz выделить группы точек, принадлежащих КК с одинаковыми номерами по оси X_T и Z_T соответственно.

7. Для каждой группы точек списков Lx и Lz найти наименьший описанный вокруг них прямоугольник, стороны которого параллельны осям X_T и Z_T системы координат текстуры соответственно рис. 2, б и рис. 2, в.

Полученные прямоугольники примем в качестве фрагментов. Полученные наборы фрагментов обозначим Gx и Gz соответственно.

8. Проанализировать сумму площадей набора фрагментов Gx и сумму площадей набора фрагментов Gz. Группу с меньшей площадью будем использовать в дальнейшем для нахождения интегрального цвета проекции пикселя.

Нахождение интегрального цвета проекции пикселя. Следует отметить, что плоскость текстуры в общем случае состоит из повторяющихся единичных текстур (рис. 2, г). Под единичной текстурой будем понимать текстуру в формате [10], ограниченную единичным квадратом.

Фрагмент может покрывать несколько единичных текстур (рис. 2, г). Область пересечения единичной текстуры с фрагментом назовем *дочерним фрагментом* (ДФ). На (рис. 2, г) один из ДФ выделен жирной линией.

С целью исключения операций масштабирования в процессе нахождения интегрального цвета ДФ переведем координаты вершин ДФ на плоскости текстуры из с/к $X_T Z_T$ в нормированные двумерные координаты точек на единичной текстуре в с/к $X_N Z_N$, $P_N \in [0,1)^2$ (рис. 2, г).

Взвешенный цвет ДФ вычисляется по формуле:

$$C^*_{\partial\phi\, ij} = C_{\partial\phi\, ij} \times S_{\partial\phi\, ij}, \qquad (3)$$

где $C^*_{\partial\phi\, ij}$ – взвешенный цвет i-го ДФ j-го фрагмента; $C_{\partial\phi\, ij}$ – интегральный цвет i-го ДФ j-го фрагмента; $S_{\partial\phi\, ij}$ – площадь i-го ДФ j-го фрагмента.

Нахождение $C^*_{\partial\phi\, ij}$ ДФ осуществляется путем суммирования с проходом по дереву подготовленных текстур в формате, описанном в [10]. Алгоритмы нахождения взвешенного цвета ДФ с проходом по дереву будут описаны в следующей статье этого цикла.

Взвешенный цвет фрагмента является суммой взвешенных цветов его ДФ. Для нахождения интегрального цвета проекции пикселя необходимо сумму взвешенных цветов фрагментов разделить на сумму их площадей.

$$C = \frac{\sum_j \sum_i^{N_j} C^*_{\partial\phi\, ij}}{\sum_j S_{\phi j}}, \qquad (4)$$

где N_j – количество ДФ j-го фрагмента; $S_{\phi j}$ – площадь j-го фрагмента; C – интегральный цвет аппроксимации проекции пикселя.

Вычисление уровня *k* разбиения на фрагменты. Глубина дерева текстур в формате [10] определяется необходимым уровнем запоминания

детализации с учетом углового разрешения системы визуализации. С целью сокращения объема вычислений, необходимого для нахождения интегрального цвета ПП, при сохранении заданной погрешности аппроксимации проекции пикселя целесообразно ограничиться некоторым уровнем детализации, соответствующим данному уровню погрешности. Для этого нужно соответствующим образом выбрать уровень k разбиения на фрагменты. Рассмотрим задачу выбора уровня k.

Для нахождения уровня k необходимо иметь такие данные: периметр ПП в с/к $X_T Z_T$, площадь ПП, максимально допустимую относительную погрешность δ аппроксимации проекции пикселя. Точное значение относительной погрешности аппроксимации ПП с помощью фрагментов получается из соотношения:

$$\delta = \frac{S_{\text{фр}} - S_{\text{ПП}}}{S_{\text{ПП}}}, \tag{5}$$

где $S_{\text{фр}} = \sum_j S_{\text{ф}j}$ – суммарная площадь фрагментов; $S_{\text{ПП}}$ – площадь ПП.

Определим количество N классификационных квадратов уровня k, через которые проходит периметр ПП

$$N \approx \frac{P_{\text{ПП}}}{l^{(k)}}, \tag{6}$$

где $l^{(k)}$ – длина стороны классификационного квадрата уровня k; $P_{\text{ПП}}$ – периметр ПП.

Абсолютную погрешность ΔS аппроксимации ПП примем равной половине суммарной площади классификационных квадратов, через которые проходит граница ПП:

$$\Delta S \approx \frac{N \cdot (l^{(k)})^2}{2}. \tag{7}$$

Тогда приближенное значение относительной погрешности (5) оценим из соотношения:

$$\delta = \frac{\Delta S}{S_{\text{ПП}}} = \frac{(l^{(k)})^2 \cdot P_{\text{ПП}}}{l^{(k)} \cdot 2 \cdot S_{\text{ПП}}} = \frac{l^{(k)} \cdot P_{\text{ПП}}}{2 \cdot S_{\text{ПП}}}. \tag{8}$$

Так как согласно [10] $l^{(k)} = 2^{-k}$ и

$$k = \left\lceil -\log_2 \frac{\delta \cdot S_{\text{ПП}}}{P_{\text{ПП}}} - 1 \right\rceil. \tag{9}$$

В нашем случае по заданной δ вычисляется номер уровня k разбиения на фрагменты.

Выводы. Предложен подход к нанесению текстуры на поверхность 3*D*-объекта произвольной формы с одновременным устранением алиасинга методом обратного трассирования. Данный подход позволяет получить аппроксимацию проекции пикселя в системе координат текстуры и привести ее к форме, удобной для вычисления интегрального цвета с использованием текстур, формат которых описан в [10]. Достоинства данного подхода к нанесению текстур: устранение алиасинга, малые вычислительные затраты, требуемая точность вычисления аппроксимации проекции пикселя, отсутствие размытия деталей рисунка текстуры. Дальнейшие исследования связаны с разработкой алгоритмов вычисления интегрального цвета ДФ с проходом по дереву текстур в формате [10] и уменьшения их вычислительной сложности.

Список литературы: **1.** *Foley J.D., van Dam A., Feiner S.K., Hughes J.F.* Computer Graphics (principles and practice) by Addison-Wesley Publishing Company, Inc., 1996. – 1175 p. **2.** *Williams L.,* Pyramidal Parametrics, S1GGRAPH 83, 1983. – P. 1–11. **3.** *Heckbert, P.S.,* Filtering by Repeated Integration, SIGGRAPH 86, 1986. – P. 315–321. **4.** *Samet H., Webber R.E.* Hierarchical Data Structures and Algorithms for Computer Graphics, Part I: Fundamentals, CG&A, May 1988. – P. 48–68. **5**. *Stefan Horbelt, Philippe Th´evenaz, Michael Unser,* Texture Mapping by Successive Refinement Proceedings of the 2000 IEEE International Conference on Image Processing (ICIP'00), Vancouver BC, Canada, September 10–13, 2000. – Vol. II. – P. 307–310. **6**. *Borgeat L., Godin G., Blais F., Massicotte P., Lahanier C.* Gold: interactive display of huge colored and textured models // ACM Trans. Graph. 24. – № 3 (2005). – P. 869–877. **7.** *Hwa L M., Duchaineau M.A., Joy K.I.* Adaptive 4-8 texture hierarchies / In Proc. Visualization. – 2004. – P. 219 – 226. **8.** *Crow, F.C.* Summed-Area Tables for Texture Mapping, SIGGRAPH 84, 1984. – P. 207–212. **9.** *John C. Hart, Nate Carr, Masaki Kameya, Stephen A. Tibbitts, Terrance J. Coleman.* Antialiased parameterized solid texturing simplified for consumer-level hardware implementation, 1999 SIGGRAPH / Eurographics Workshop on Graphics Hardware, Aug., 1999. – P. 45–53. **10.** *Гусятин В.М., Чаговец Я.В., Кожушко Д.Г.* Упаковка векторных текстур в задачах синтеза изображений для систем визуализации. Вісник НТУ "ХПІ". Серія "Інформатика і моделювання". – 2005. – № 56. – С. 9–16.

Поступила в редакцию 10.10.2007